

ED-CEC: IMPROVING RARE WORD RECOGNITION USING ASR POSTPROCESSING BASED ON ERROR DETECTION AND CONTEXT-AWARE ERROR CORRECTION

Jiajun He^{*}, Zekun Yang[‡], Tomoki Toda[‡]

^{*} Graduate School of Informatics, Nagoya University, Japan

[‡] Information Technology Center, Nagoya University, Japan

ABSTRACT

Automatic speech recognition (ASR) systems often encounter difficulties in accurately recognizing rare words, leading to errors that can have a negative impact on downstream tasks such as keyword spotting, intent detection, and text summarization. To address this challenge, we present a novel ASR postprocessing method that focuses on improving the recognition of rare words through error detection and context-aware error correction. Our method optimizes the decoding process by targeting only the predicted error positions, minimizing unnecessary computations. Moreover, we leverage a rare word list to provide additional contextual knowledge, enabling the model to better correct rare words. Experimental results across five datasets demonstrate that our proposed method achieves significantly lower word error rates (WERs) than previous approaches while maintaining a reasonable inference speed. Furthermore, our approach exhibits promising robustness across different ASR systems.

Index Terms— automatic speech recognition, rare words, error detection, context-aware error correction, rare word list

1. INTRODUCTION

Automatic speech recognition (ASR) technology has made considerable progress in recent years, enabling machines to transcribe speech with marked accuracy [1, 2]. However, even with state-of-the-art (SOTA) ASR systems, there remains a persistent challenge in accurately recognizing rare words, such as named entities, technical terms, and specific names [3]. These rare words are often misrecognized as similar-sounding words in the recognition lexicon, resulting in errors that significantly degrade the overall transcription quality [4]. Such errors can have a substantial impact on downstream tasks such as video summarization [5] and named entity recognition [6, 7]. Consequently, improving the recognition of rare words has become a crucial objective in enhancing ASR performance.

To tackle the challenge of rare word recognition in ASR, several techniques have been proposed. These techniques primarily involve incorporating contextual knowledge into the ASR system [3, 8–10] and integrating an additional language model (LM) into the decoding phase to bias recognition results towards contextual knowledge [11–14]. In these approaches, contextual knowledge is typically represented by a list of words or phrases, known as contextual items, that are likely to appear in a given context. Various resources, such as lecture video slides, meeting minutes, and a user’s contact book, can be utilized to construct the rare word list [15, 16]. However, these aforementioned approaches have certain limitations. On one hand, the method of incorporating contextual knowledge into the ASR system can be computationally expensive during both training and inference, and it may require significant modifications to the original ASR models’ structure [8]. Moreover, this approach may not effectively handle a large rare word list. On the other hand, the method

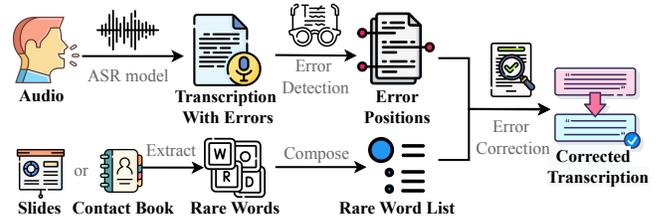


Fig. 1. Pipeline of our proposed method.

of integrating an additional LM into the decoding phase necessitates careful weight tuning in different scenarios.

To address the limitations of previous approaches, ASR error correction (AEC) has been proposed [17–21]. An AEC model is designed to be an independent model that does not alter the structure of the original ASR model, ensuring no risk of performance degradation. This characteristic makes it highly convenient to apply the AEC model across various domains, as it can be easily integrated by replacing the existing AEC model without the need for retraining the original ASR model [19]. Wang et al. [19] integrated contextual knowledge into an error correction model through a context encoder, which corrects the ASR output from scratch. However, this process raised significant concerns regarding inference speed. Interestingly, it was observed that the majority of words were identical between the ASR output and the ground truth. Hence, Yang et al. [20] proposed the use of an operation predictor to constrain the decoding process, resulting in a notable improvement in inference speed while retaining the capability to correct certain errors. However, owing to the lack of contextual knowledge integration, this approach could not effectively correct rare words.

In this paper, we propose a novel method based on error detection and context-aware error correction (ED-CEC) to address the challenges associated with inference speed and rare word correction, as shown in Fig. 1. The rare word list used in the error correction module can be obtained from various sources such as slide texts or a contact book. The contributions of this paper are summarized below:

- We propose a method to correct rare words based on ASR results. Our model includes an error detection module, which identifies incorrect positions and decodes only those positions to increase inference speed, and a context-aware error correction module, which corrects rare words by selecting relevant contextual items from a rare word list.
- We conduct experiments on five datasets. The results demonstrate that our model achieves a relative word error rate reduction (WERR) ranging from 15.6% to 38.17% compared with the original ASR output and the average relative improvement in biased word error rate (B-WER) is 46.68%. In addition, the proposed method achieves an inference speed of 2.8 – 6.0 times higher than the previous SOTA model.

applied to obtain the decoder layer output, where the query input Q is the decoder input. Both the key input K and the value input V are the output of the BERT encoder of the model input I :

$$Q = E_{t,k}^Z, K = E^I, V = E^I \quad (4)$$

$$O_{t+1,k}^{gen} = \text{Transformer}_{\text{Decoder}}(Q, K, V), \quad (5)$$

where $O_{t+1,k}^{gen}$ is the decoder layer output. Finally, the generation output is calculated as:

$$p_{t+1,k}^{gen} = \text{Softmax}(\text{FC}(O_{t+1,k}^{gen})). \quad (6)$$

To dynamically choose between selecting tokens from the rare word list and generating new tokens, we also introduce a novel contextual mechanism. We use the contextual mechanism comprising a contextual encoder and a contextual decoder, with the contextual decoder consisting of context attention and context-item attention. The following are the detailed descriptions:

Contextual Encoder. We store l contextual items in our rare word list. The j^{th} contextual item, denoted as $c_j = (c_j^1, \dots, c_j^u)$, is represented by WordPiece tokenization mentioned above, where u indicates the number of tokens in the respective contextual item. To optimize the model size and increase inference speed, we adopt parameter sharing between the BERT encoder and the contextual encoder. As a result, we utilize the identical BERT encoder to obtain the hidden representations of the contextual items:

$$E^C = \text{BERT}(\text{TE}(C) + \text{PE}(C)), \quad (7)$$

where C is the rare word list mentioned in Section 2.1. TE and PE are the token embedding and position embedding mentioned above. $E^C = (e_1^C, \dots, e_l^C)$ is the output of the contextual encoder.

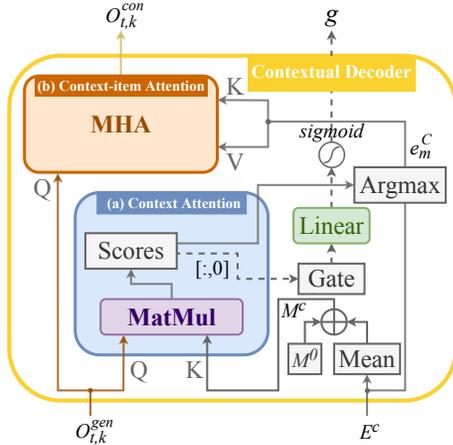


Fig. 3. Overall architecture of contextual decoder.

Contextual Decoder. The model diagram of the contextual decoder is shown in Fig. 3. We compute the mean of each encoded contextual item and then expand these tokens with a learnable dummy token $\langle no\text{-}context \rangle$, which is utilized later to determine situations where there is no relevant knowledge stored in the rare word list:

$$M^C = M^0 \oplus \text{mean}(E^C), \quad (8)$$

where M^0 is the hidden representation of the learned dummy token $\langle no\text{-}context \rangle$ and mean is the mean operation on each $e_j^C \in E^C$, $j \in \{1, \dots, l\}$. M^C can be interpreted as summary tokens of contextual items.

At step t , the contextual decoder begins with a context attention layer that identifies the availability and location of a relevant contextual item in the rare word list by computing similarity scores, where

the query input Q_t and the key input K_t are the output of the decoder layer and the summary tokens of contextual items at step t , respectively:

$$Q_t = O_{t,k}^{gen}, K_t = M^C \quad (9)$$

$$\text{scores}_t = Q_t K_t^T \quad (10)$$

$$\text{gate}_t = \text{scores}_t[:, 0], \quad (11)$$

where $\text{scores}_t[:, 0]$, namely, gate_t are the similarity scores corresponding to the $\langle no\text{-}context \rangle$ token M^0 . We define the index of the highest similarity score for the query Q_t at step t as $m = \text{argmax}(\text{scores}_t)$. If m is non-zero, indicating the presence of relevant contextual knowledge in the rare word list, we compute the contextual output using the context-item attention layer. This layer extracts the relevant information from a specific contextual item using a multihead attention (MHA) mechanism. In the MHA layer, the query input Q_t is the output of the decoder layer, while the key input K_t and the value input V_t are both obtained from the contextual encoder of the m^{th} contextual item at step t :

$$Q_t = O_{t,k}^{gen}, K_t = e_m^C, V_t = e_m^C \quad (12)$$

$$O_{t+1,k}^{con} = \text{Softmax}\left(\frac{Q_t K_t^T}{\sqrt{d}}\right) V_t \quad (13)$$

$$p_{t,k}^{con} = \text{Softmax}(\text{FC}(O_{t,k}^{con})), \quad (14)$$

where the scaling factor \sqrt{d} is for numerical stability. $O_{t,k}^{con}$ is the output of the context-item attention layer.

Then, the predicted word is acquired by a weighted sum between the generation output $p_{t,k}^{gen}$ and the contextual output $p_{t,k}^{con}$:

$$g = \sigma(\text{FC}_{\text{dim}=1}(\text{gate}_t)) \quad (15)$$

$$P_{t,k} = g \cdot p_{t,k}^{gen} + (1 - g) \cdot p_{t,k}^{con}, \quad (16)$$

where $\text{FC}_{\text{dim}=1}$ denotes one fully connected layer with one output neuron, σ is the sigmoid function and g is the gate to make a trade-off between the chosen token from the rare word list and the generated token.

2.4. Joint Training and Completion

The learning process is optimized through two objectives that correspond to error detection and context-aware error correction.

$$\text{Loss}_d = - \sum_o \log(P(y_o | i_o)) \quad (17)$$

$$\text{Loss}_e = - \left(\sum_k \sum_t \log(P_{t,k}) + \sum_k \sum_t \log(P(\text{label}_{t,k} | \text{scores}_{t,k})) \right), \quad (18)$$

where the loss function Loss_d is the cross entropy loss for the detection network and the loss function Loss_e consists of two parts of the cross entropy loss for the context-aware correction network. Furthermore, $\text{scores}_{t,k}$ are the score outputs of the context attention layer and $\text{label}_{t,k}$ is the contextual label that contains the index of the corresponding contextual item. The two loss functions are linearly combined as the overall objective in the learning phase:

$$\text{Loss} = \gamma \cdot \text{Loss}_d + \text{Loss}_e, \quad (19)$$

where γ is the hyperparameter for adjusting the weight between Loss_d and Loss_e .

During the completion process, we convert the predicted operation labels and the generated words into a complete utterance. Specifically, as depicted in Fig. 2, we preserve the tokens labeled \mathbf{K} and remove those labeled \mathbf{D} from the inputs. We then replace the tokens labeled \mathbf{C} with the corresponding generated words.

3. EXPERIMENTAL EVALUATIONS

3.1. Experiment Settings

Our method was implemented using Python 3.7 and Pytorch 1.11.0. The model was trained and evaluated on a computer with Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz, 32GB RAM, and one NVIDIA Tesla V100 GPU.

Both the BERT encoder and the contextual encoder employed the same bert-base-uncased model [23] for initialization. The vocabulary size for word tokenization was 30522. We set the hidden size as 768, the number of attention layers as 12, and the number of attention heads as 12. The transformer decoder adopted a single-layer transformer with a hidden size of 768. We used Adam [24] as the optimizer with a batch size of 32 and set γ to 3. The initial learning rate was 0.00005. All the hyperparameters were fine-tuned on the standard validation data.

3.2. Data

To assess the effectiveness and robustness of our proposed model, we employ five datasets that utilize various ASR engines. The statistics of the datasets are shown in Table 1.

- The ATIS dataset [25] includes 8 hours of audio recordings of people making flight reservations, along with their corresponding human transcripts. ASR transcripts were generated by a LAS ASR system [26].
- The SNIPS dataset [27] is collected from the SNIPS voice assistant, focusing on natural language understanding. The Kaldi¹ ASR toolkit was used to obtain the corresponding transcripts.
- The Librispeech dataset [28] is a collection of 960 hours of audiobooks. The ESPNet [29] ASR toolkit was utilized to obtain related transcripts. We used dev-clean and test-clean as the validation and test sets, respectively.
- The MELD dataset [30] consists of more than 1400 dialogues and 13000 utterances extracted from the Friends TV series. We utilized Whisper [31] to obtain transcripts.
- The PRLVS dataset [32] comprises a complete semester course consisting of pattern recognition lecture videos accompanied by slides. The course consists of 43 videos, with a total duration of 11.4 hours. We employed SpeechBrain [33] to obtain the transcripts related to the videos.

Dataset	ATIS	SNIPS	Librispeech	MELD	PRLVS
Train	3867	13084	252691	9989	3680
Valid	967	700	2703	1109	460
Test	800	700	2620	2610	460

Table 1. Numbers of utterances in different datasets.

3.3. Rare Word List Construction

Owing to the lack of available rare word lists for the ATIS, SNIPS, Librispeech, and MELD datasets, we followed the approach proposed in [34] to construct rare word lists for each dataset. Specifically, a complete rare word list was initially compiled for the Librispeech dataset, consisting of 209.2K distinct words, by excluding the top 5,000 most common words from the Librispeech LM training corpus. Next, the rare word lists were constructed by identifying words from the reference of each utterance that were present in the complete rare word list. Additionally, a specified number of distractors (e.g., 1,000) were added to each rare word list, as determined by the experiment requirements. By utilizing this approach, we can

¹<https://github.com/kaldi-asr/kaldi>

effectively organize the rare word lists for each utterance², containing words from the complete rare word list and supplementing them with distractors. Similar methods were employed to construct rare word lists for the remaining ATIS, SNIPS, and MELD datasets.

To demonstrate the feasibility of obtaining rare word lists in practice, we focused on the PRLVS dataset. The construction process involved collecting slides for each lecture and utilizing the Tesseract 4 OCR engine³ for text extraction. Distinct word tokens were then extracted from the OCR output files. Among these tokens, only those belonging to the complete rare word list and appearing fewer than 15 times in the PRLVS train set were included in the lecture-specific rare word list. These rare word lists were subsequently applied for the context-aware correction of all utterances within the corresponding lectures [35].

3.4. Baselines and Metrics

We evaluate the error correction performance of our proposed method, as well as four baselines:

- Original denotes the original ASR output.
- SC.BART [36] has demonstrated superior performance in ASR error correction tasks, achieving SOTA results.
- distillBART [37] is a distilled version of the BART large model.
- ConstDecoder_{trans} [20] is a constrained decoding method designed to improve the inference speed of ASR error correction while preserving a certain level of error correction performance.

In addition, we use the following four evaluation metrics to assess the performance:

- WER is the overall word error rate (WER) on all words.
- WERR quantifies the WER reduction across all words.
- U-WER calculates the unbiased WER on words not included in the rare word list.
- B-WER computes the biased WER on words present in the rare word list.

In the case of insertion errors, if the inserted word is found in the rare word list, it will contribute to B-WER; otherwise, it will be considered for U-WER. The objective of contextualization is to improve B-WER while minimizing any significant degradation in the U-WER [34].

Method	ATIS	SNIPS	Librispeech	MELD	PRLVS
SC.BART	90.30	75.30	152.93	25.70	103.62
distillBART	45.55	41.55	73.29	11.99	57.60
ConstDecoder _{trans}	25.61	26.66	17.10	4.23	18.29
ED-CEC (Proposed)	32.59	31.87	25.48	5.71	22.86
vs SC.BART	2.8×	2.4×	6.0×	4.5×	4.5×
vs distillBART	1.4×	1.3×	2.9×	2.1×	2.51×
vs ConstDecoder _{trans}	0.8×	0.8×	0.7×	0.7×	0.8×

Table 2. Average inference time in milliseconds (ms).

3.5. Results and Analysis

Tables 2 and 3 provide evidence that our model achieves significant improvements in inference speed and WER results compared with the previous SOTA model on all five datasets when the rare word list size is set to 100. Compared with the original ASR output, our model achieves a marked WERR, ranging from 15.6% to 38.17%. The average relative improvement in B-WER is 46.68%. Furthermore, our model demonstrates considerable gains in inference speed, being 2.4 to 6.0 times higher than the previous SOTA model. This proves that our model achieves a good tradeoff between inference speed

²https://github.com/facebookresearch/fbai-speech/tree/master/is21_deep_bias

³<https://github.com/tesseract-ocr/tesseract>

Method	ATIS	SNIPS	Librispeech	MELD	PRLVS
	WER/WERR (U-WER/B-WER)	WER/WERR (U-WER/B-WER)	WER/WERR (U-WER/B-WER)	WER/WERR (U-WER/B-WER)	WER/WERR (U-WER/B-WER)
Original	30.65/- (20.58/ 87.78)	45.73/- (34.20/99.64)	6.75/ - (3.13/30.26)	31.08/ - (25.31/ 74.62)	18.66/ - (10.66/ 47.24)
SC_BART	21.47/29.95 (14.63/49.25)	30.35/33.63 (21.86/70.25)	5.78/14.37 (3.80/18.45)	29.51/ 5.05 (24.25/67.73)	15.14/ 18.86 (10.43/27.67)
distillBART	26.51/13.51 (18.54/74.67)	33.28/27.23 (24.08/76.43)	6.36/5.78 (3.88/23.49)	30.76/1.03 (25.06/73.89)	17.98/ 3.64 (10.64/ 43.57)
ConstDecoder _{trans}	21.74/29.07 (14.78/50.57)	30.98/32.25 (22.09/71.43)	5.89/12.74 (3.68/19.94)	29.98/ 3.54 (24.70/69.42)	15.31/ 17.95 (10.95/28.33)
ED-CEC (Proposed)	18.95/38.17 (14.88/38.38)	28.57/37.52 (21.47/62.79)	5.08/24.74 (3.77/12.38)	26.23/15.60 (21.31/56.02)	13.17/ 29.42 (10.01/ 20.72)

Table 3. Measurements of error correction performance on five datasets.

and WER. Additionally, the performance improvements across five different ASR systems prove the robustness of our model.

We also experimented on the PRLVS dataset with varying rare word list sizes, created by augmenting the rare word lists with distractors, ranging from 100 to 1000 contextual items. The best WER results were observed with a rare word list size of 100, as shown in Fig. 4. Increasing the rare word list size to 1000 showed a minor upward trend in WER, possibly due to false positives. Importantly, an empty rare word list resulted in a significant WER increase, highlighting the model’s reliance on contextual items for rare word correction. Furthermore, we conducted an “anti-context” experiment, employing a rare word list only containing 100 unrelated distractors. In this case, the WER was 15.45%. Thus, our approach yields optimal results when combined with a small number of relevant rare words that the model should prioritize.

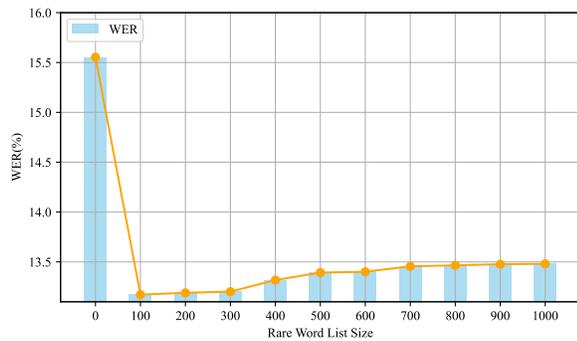


Fig. 4. WER results of PRLVS with different rare word list sizes.

Fig. 5 shows a specific example of correcting rare words “musulmans” and “giaours”. The ASR refers to the original ASR output. In (a), decoding is performed with an empty rare word list, whereas in (b), decoding is performed with a rare word list of size 10 containing the rare words “musulmans” and “giaour”. The GT denotes the ground truth transcript. When the rare word list is empty, the heat map of the gate (the weights between the original transformer decoder and the contextual decoder) indicates a stronger preference towards the output of the original transformer decoder. The generated words “mumen” and “gas” significantly deviate from the ground truth. However, when the rare word list contains the target rare words, the gate tends to favor the output of the contextual decoder, and the heat map of the rare word list shows that the model correctly selects the positions of the rare words.

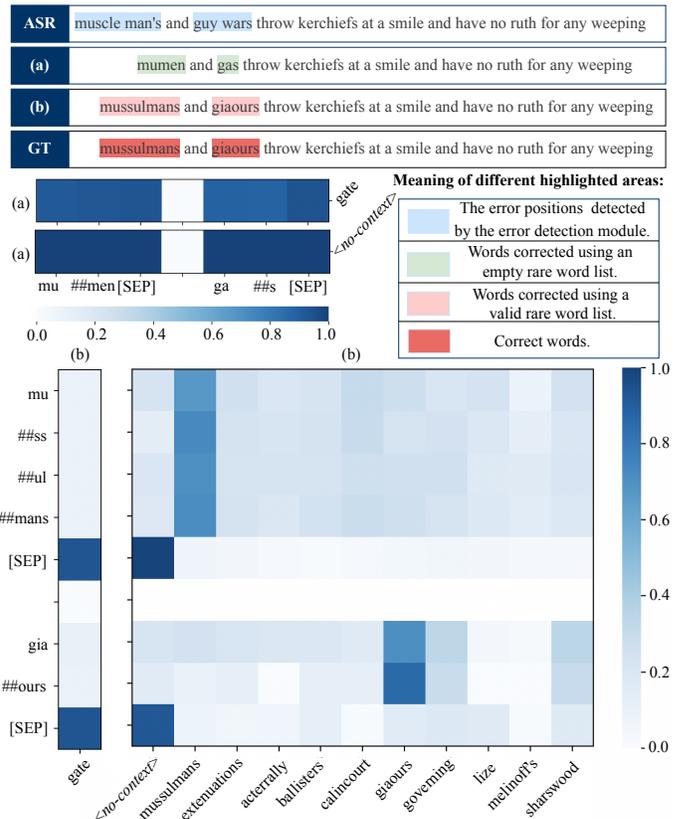


Fig. 5. Example for correcting rare words.

4. CONCLUSION AND FUTURE WORK

In this paper, we propose a fast and efficient contextual ASR error correction method that incorporates two main modules: an error detection module and a context-aware error correction module. This approach ensures both a high inference speed and the accurate correction of rare word errors in the ASR output. Experimental results on five datasets show the effectiveness and robustness of our model. In the future, we plan to extend our model by adding an additional phoneme encoder to recognize error patterns at the phoneme level, which will enable us to better bias rare words for correction.

Acknowledgement. This work was supported in part by JST CREST Grant Number JPMJCR22D1, Japan, and a project, JPNP20-006, commissioned by NEDO.

References

- [1] Jinyu Li et al., “Recent advances in end-to-end automatic speech recognition,” *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.
- [2] Zhong Meng et al., “Jeit: Joint end-to-end model and internal language model training for speech recognition,” *Proc. ICASSP*, pp. 1–5, 2023.
- [3] Christian Huber et al., “Instant one-shot word-learning for context-specific neural sequence-to-sequence speech recognition,” *Proc. ASRU*, pp. 1–7, 2021.
- [4] Dhanush Bekal et al., “Remember the context! asr slot error correction through memorization,” *Proc. ASRU*, pp. 236–243, 2021.
- [5] Roshan Sharma et al., “End-to-end speech summarization using restricted self-attention,” *Proc. ICASSP*, pp. 8072–8076, 2022.
- [6] Ido Cohn et al., “Audio de-identification: A new entity recognition task,” *Proc. NAACL-HLT*, pp. 197–204, 2019.
- [7] Guillaume Baril et al., “Named entity recognition for audio de-identification,” *Proc. IJCNN*, pp. 1–8, 2022.
- [8] Golan Pundak et al., “Deep context: end-to-end contextual speech recognition,” *Proc. SLT*, pp. 418–425, 2018.
- [9] Minglun Han et al., “Improving end-to-end contextual speech recognition with fine-grained contextual knowledge selection,” *Proc. ICASSP*, pp. 8532–8536, 2022.
- [10] Yufei Liu et al., “Internal language model estimation through explicit context vector learning for attention-based encoder-decoder asr,” *Proc. Interspeech*, pp. 1666–1670, 2022.
- [11] Ian Williams et al., “Contextual speech recognition in end-to-end neural network systems using beam search,” *Proc. Interspeech*, pp. 2227–2231, 2018.
- [12] Duc Le et al., “Deep shallow fusion for rnn-t personalization,” *Proc. SLT*, pp. 251–257, 2021.
- [13] Wei Zhou et al., “On language model integration for rnn transducer based speech recognition,” *Proc. ICASSP*, pp. 8407–8411, 2022.
- [14] Xie Chen et al., “Factorized neural transducer for efficient language model adaptation,” *Proc. ICASSP*, pp. 8132–8136, 2022.
- [15] Joao Miranda et al., “Improving asr by integrating lecture audio and slides,” *Proc. ICASSP*, pp. 8131–8135, 2013.
- [16] Yuya Akita et al., “Language model adaptation for academic lectures using character recognition result of presentation slides,” *Proc. ICASSP*, pp. 5431–5435, 2015.
- [17] Ziji Zhang et al., “Patcorrect: Non-autoregressive phoneme-augmented transformer for asr error correction,” *Proc. Interspeech*, pp. 3904–3908, 2023.
- [18] Samrat Dutta et al., “Error correction in asr using sequence-to-sequence models,” *arXiv:2202.01157*, 2022.
- [19] Xiaoqiang Wang et al., “Towards contextual spelling correction for customization of end-to-end speech recognition systems,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 3089–3097, 2022.
- [20] Jingyuan Yang et al., “Asr error correction with constrained decoding on operation prediction,” *Proc. Interspeech*, p. 3874–3878, 2022.
- [21] Binghui Lin et al., “Multi-modal asr error correction with joint asr error detection,” *Proc. ICASSP*, pp. 1–5, 2023.
- [22] Yonghui Wu et al., “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *arXiv:1609.08144*, 2016.
- [23] Jacob Devlin et al., “Bert: Pre-training of deep bidirectional transformers for language understanding,” *Proc. NAACL-HLT*, p. 4171–4186, 2019.
- [24] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *Proc. ICLR*, pp. 1–11, 2015.
- [25] Sundararaman et al., “Phoneme-bert: Joint language modelling of phoneme sequence and asr transcript,” *Proc. Interspeech*, pp. 3236–3240, 2021.
- [26] William Chan et al., “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” *Proc. ICASSP*, pp. 4960–4964, 2016.
- [27] Chao-Wei Huang and Yun-Nung Chen, “Learning asr-robust contextualized embeddings for spoken language understanding,” *Proc. ICASSP*, pp. 8009–8013, 2020.
- [28] Vassil Panayotov et al., “Librispeech: An asr corpus based on public domain audio books,” *Proc. ICASSP*, pp. 5206–5210, 2015.
- [29] Shinji Watanabe et al., “Espnet: End-to-end speech processing toolkit,” *Proc. Interspeech*, pp. 2207–2211, 2018.
- [30] Soujanya Poria et al., “Meld: A multimodal multi-party dataset for emotion recognition in conversations,” *Proc. ACL*, pp. 527–536, 2018.
- [31] Alec Radford et al., “Robust speech recognition via large-scale weak supervision,” *arXiv:2212.04356*, 2022.
- [32] Abner Hernandez and Seung Hee Yang, “Multimodal corpus analysis of autoblog 2020: lecture videos in machine learning,” *Proc. SPECOM*, pp. 262–270, 2021.
- [33] Mirco Ravanelli et al., “Speechbrain: A general-purpose speech toolkit,” *arXiv:2106.04624*, 2021.
- [34] Duc Le et al., “Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion,” *Proc. Interspeech*, pp. 1772–1776, 2021.
- [35] Guangzhi Sun et al., “Tree-constrained pointer generator with graph neural network encodings for contextual speech recognition,” *Proc. Interspeech*, pp. 2043–2047, 2022.
- [36] Yun Zhao et al., “Bart based semantic correction for mandarin automatic speech recognition system,” *Proc. Interspeech*, pp. 2017–2021, 2021.
- [37] Sam Shleifer and Alexander M Rush, “Pre-trained summarization distillation,” *arXiv:2010.13002*, 2020.